บทที่ 2

ทฤษฎีทางด้านฮาร์ดแวร์

2.1 สเตปเปอร์มอเตอร์ (Stepper motor)

สเตปเปอร์มอเตอร์คือมอเตอร์ชนิดหนึ่งที่ทำการหมุนเป็นขั้นๆทีละน้อยตามเส้นรอบวงและหยุด แตกต่างกับมอเตอร์ทั่วๆไปซึ่งจะหมุนไปเรื่อยๆตลอดเวลา เราสามารถควบคุมการหมุนของสเตปเปอร์มอเตอร์ได้อย่างละเอียดโดยใช้คอมพิวเตอร์ ไมโครคอนโทรลเลอร์ หรือแม้กระทั่งวงจรลอจิก สเตปเปอร์มอเตอร์สามารถใช้งานที่ต้องการความละเอียดต่างๆเช่น ควบคุมการหมุนกระดาษในพรินเตอร์ หรือใช้ในการเคลื่อนที่หัวอ่านเพื่ออ่านและเขียนของฟลอปปี้ดิสก์เป็นต้น สเตปเปอร์มอเตอร์ใช้งานในลักษณะระบบเปิด (Open Loop System) คือสเตปเปอร์มอเตอร์สามารถทำงานได้โดยไม่ต้องมีการป้อนค่า พารามิเตอร์กลับมาที่วงจรควบคุม (Feedback) แต่ในงานเฉพาะทางที่ต้องการความแน่นอนของตำแหน่งของมอเตอร์ ก็สามารถใช้เซนเซอร์ตรวจจับตำแหน่งร่วมกับสเตปเปอร์มอเตอร์ได้โดยง่าย

สเตปเปอร์มอเตอร์เป็นอุปกรณ์ที่ทำหน้าที่แปลงสัญญาณข้อมูลดิจิตอล ไปเป็นการเคลื่อนที่เชิงกลอย่างเป็นสัดส่วน โดยแกนหรือโรเตอร์ของมอเตอร์ชนิดนี้จะถกควบคุมให้หมุนไปเป็นลำดับขั้น

ประโยชน์จากการใช้งานสเตปเปอร์มอเตอร์ที่มีการควบคุมโดยใช้สัญญาณดิจิตอล มีความถูกต้องและแม่นยำ และสามารถเปลี่ยนตำแหน่งของโหลดได้อย่างรวดเร็ว เนื่องจากแต่ละ สัญญาณเข้ามาหนึ่งพัลซ์ จะทำให้สเตปเปอร์มอเตอร์เคลื่องที่ไปหนึ่งสเตปอย่างเที่ยงตรง

เมื่อพิจารณาในแง่ทางกลจะพบว่าสเตปเปอร์มอเตอร์ที่มีความเที่ยงตรงที่ยอมรับได้ซึ่งในการเปลี่ยนตำแหน่งอย่างง่ายอาจใช้สวิตช์ ในการควบคุมมอเตอร์ นั่นคือสร้างวงจรควบคุมได้ง่าย ถ้าต้องการเพิ่มประสิทธิภาพของส่วนควบคุม ให้ดีขึ้น อาจใช้ไอซีที่มีความเร็วสูง เนื่องจากประกอบด้วยมอฟเฟตอยู่ภายใน จึงได้กำลังงานสูง และต้นทุนต่ำ ความสะดวกในการใช้งานนี้เองจึงทำให้นำไปสู่การใช้งานอย่างแพร่หลาย

การได้รับประโยชน์จากสเตปเปอร์มอเตอร์อย่างเต็มที่นั้นขึ้นอยู่กับการขับอย่างถูกตองเหมาะสม ซึ่งภาคขับนี้จะต้องประกอบไปด้วย แหล่งจ่ายไฟฟ้ากระแสตรง อิเลกทรอนิกส์สวิตซ์ โดยอาจจะเป็นทรานซิสเตอร์หรือมอสเฟส และแหล่งจ่ายสัญญาณข้อมูลแบบดิจิตอล ซึ่งมีลักษณะเป็นพัลซ์ เพื่อใช้ควบคุมทิศทางการหมุนของมอเตอร์

ในปัจจุบันสเตปเปอร์มอเตอร์ที่นิยมใช้มีอยู่ 2 ชนิดด้วยกันคือแบบแม่เหล็กถาวร และแบบที่สามารถเปลี่ยนความเข้มสนามแม่เหล็กได้ ซึ่งแบบแม่เหล็กถาวรจะสะดวกในการใช้และควบคุมมากกว่า แบ่งออกเป็น 2 แบบด้วยกันคือ

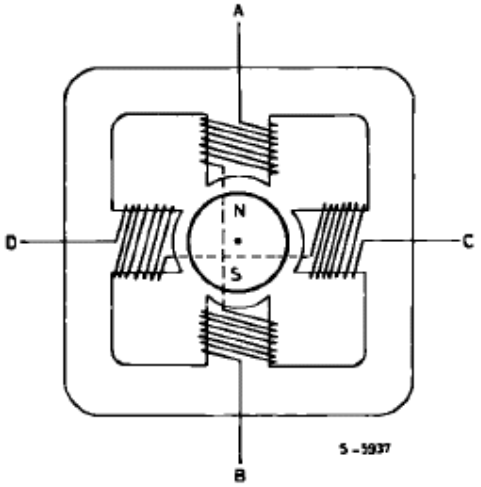
- ไบโพล่าร์ (Bipolar)
- ยูนิโพล่าร์ (Unipolar)

ซึ่งทั้งสองชนิดนี้มีความแตกต่างกันที่ขั้วของมอเตอร์ แบบยูนิโพล่าร์จะทำให้เกิดแรงบิดน้อยกว่าแบบไบโพล่าร์ ความแตกต่างทางกายภาพของสเตปเปอร์มอเตอร์ทั้งสองแบบคือสายไฟที่ต่อมาจากตัวมอเตอร์ แบบไบโพล่าร์จะมี 4 สาย ส่วนแบบยูนิโพล่าร์จะมี 5-6 สาย

2.1.1 ชนิดของสเตปเปอร์มอเตอร์

2.1.1.1 สเตปเปอร์มอเตอร์แบบไบโพล่าร์

เป็นสเตปเปอร์มอเตอร์ที่มีแกนเป็นแม่เหล็กถาวรและมีคอยล์อยู่ 4 ชุดด้วยกัน ทำงานโดยใช้กระแสไบไดเรกชันแนล (Bidirectional) ส่งกระแสผ่านจาก AB/CD/BA/DC เพื่อให้มอเตอร์หมุนไปหนึ่งเฟสแบบฟูลสเต็ป (Full step / Wave drive) สัญลักษณ์ AB คือส่งกระแสจากขั้ว A ไปขั้ว B และ BA เป็นการส่งกระแสในทิศทางตรงกันข้าม



รูปที่ 2-1 สเตปเปอร์มอเตอร์แบบไบโพล่าร์

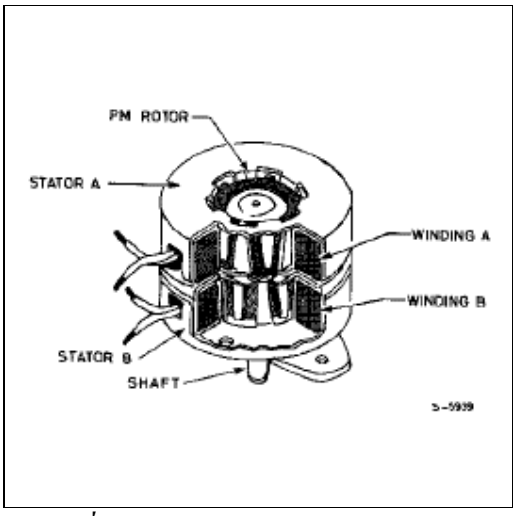
ทิศทางของกระแสไฟฟ้าที่ป้อนเข้าสู่สเตปเปอร์มอเตอร์ ถูกควบคุมโดยการทำงานของ อิเลกทรอนิกส์สวิตซ์ ดังนั้นสเตปเปอร์มอเตอร์จะหมุนไปหนึ่งช่วงในแต่ละอินพุตพัลซ์ที่ป้อนเข้าสู่วงจรอิเลกทรอนิกส์สวิตซ์ ซึ่งขนาดที่หมุนไปจะขึ้นอยู่กับการออกแบบสเตปเปอร์มอเตอร์ โดยมีค่าตั้งแต่ 1.8 องศา ถึง 15 องศา ถ้าส่งสัญญาณไปยังวงจรสวิตซ์ซึ่ง 24 พัลซ์ โดยมีค่าหนึ่งของช่วงการหมุนเท่ากับ 15 องศา สเตปเปอร์มอเตอร์ก็จะหมุนไปครบหนึ่งรอบพอดี ขึ้นอยู่กับอัตราการจ่ายสัญญาณพัลซ์ควบคุม โดยสัญญาณนี้อาจถูกผลิตโดยวงจรกำเนิดสัญญาณที่ปรับความถี่ได้ หรือจากไอซีควบคุม

2.1.1.2 สเตปเปอร์มอเตอร์แบบยูนิโพล่าร์

เป็นสเตปเปอร์มอเตอร์ที่มีแกนเป็นแม่เหล็กถาวรเช่นเดียวกับไบโพล่าร์ที่ได้กล่าวไปข้างต้น เพียงแต่ใช้การหมุนกลับของสเตเตอร์ ฟลักซ์ แทนที่จะใช้การขับแบบไบไดเรคชั่น

การขับมอเตอร์ชนิดนี้ใช้วิธีการเดียวกับไบโพล่าร์เพียงแต่นำเอาขั้นตอนแบบยูนิโพล่าร์เข้ามาแทน บริดจ์ ไดรเวอร์ โดยขั้นตอนแบบยูนิโพล่าร์นี้คือการนำทรานซิสเตอร์แบบดาลิงตัน สี่ตัวมาใช้ และสเตปเปอร์มอเตอร์แบบยูนิโพล่าร์ยังมีราคาแพงกว่าเนื่องมีขดลวดมากกว่าปืนเท่าตัว และยังให้ทอร์กต่ำกว่า

เนื่องจากขดลวดทำมาจากเส้นลวดที่มีขนาดบางกว่า ในอดีตสเตปเปอร์มอเตอร์แบบยูนิโพล่าร์เป็นที่นิยม เนื่องจากใช้บังคับได้ง่าย แต่ในปัจจุบันมีการคิดค้นไอซี L298N ที่ใช้ในการควบคุม สเตปเปอร์มอเตอร์แบบไบโพล่าร์โดยเฉพาะ ทำให้การควบคุมสเตปเปอร์มอเตอร์แบบไบโพล่าร์ง่ายกว่า ในปัจจุบันสเตปเปอร์มอเตอร์แบบไบโพล่าร์ จึงเป็นที่นิยมใช้มากกว่า



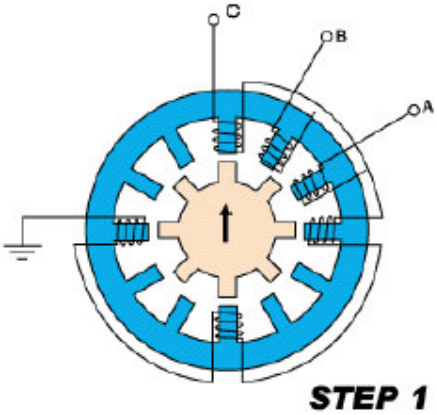
รูปที่ 2-2 สเตปเปอร์มอเตอร์แบบยูนิโพล่าร์

2.1.2 วิธีใช้สเตปเปอร์มอเตอร์แบบไบโพล่าร์

การทำงานของสเตปเปอร์มอเตอร์แบบไบโพล่าร์ ตัวสเตเตอร์และโรเตอร์ที่สร้างด้วยเหล็กผสมซิลิคอน โดยที่ตัวโรเตอร์จะไม่ถูกพันด้วยคอยล์ จะเป็นส่วนที่ถูกพันด้วยคอลย์จะเป็นส่วนของสเตเตอร์เท่านั้น ดังนั้นเมื่อเราจ่ายกระแสไฟฟ้าผ่านคอยล์ จึงทำให้สเตเตอร์มีคุณสมบัติเป็นแม่เหล็ก

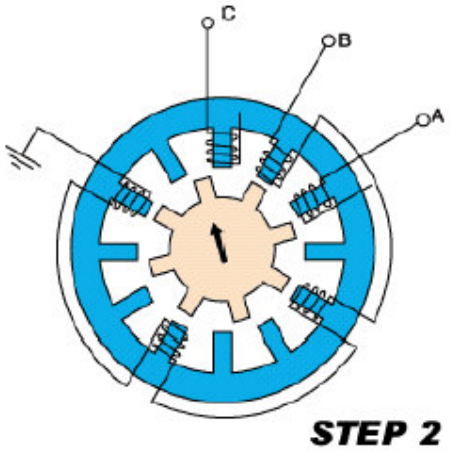
2.1.2.1 การควบคุมสเตปเปอร์มอเตอร์แบบเวฟ (Wave or Half Step Drive)

การควบคุมมอเตอร์แบบเวฟง่ายต่อการใช้งาน ใช้การส่งกระแสเข้าไปที่แต่ละขาของ สเตเตอร์แต่ละตัวเรียงลำดับกันไปเท่านั้น วิธีนี้นอกจากจะเร็วแล้วยังกินไฟต่ำอีกด้วย แต่ก็จะมีแรงบิดไม่มากนัก



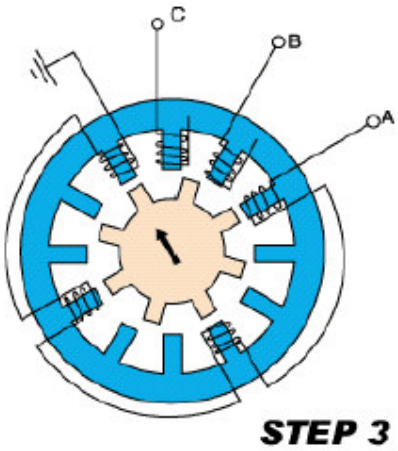
รูปที่ 2-3 การทำงานของสเตปเปอร์มอเตอร์ ขั้นที่1

ในขั้นแรกจะจ่ายกระแสให้แก่เฟส C ดังนั้นซี่ของโรเตอร์ก็จะถูกดูดเข้าหาเฟส C



รูปที่ 2-4 การทำงานของสเตปเปอร์มอเตอร์ ขั้นที่2

สเตปต่อไปหยุดจ่ายกระแสให้กับเฟส C และจ่ายให้กับเฟส B แทน ซี่ของโรเตอร์อันถัดไปที่อยู่ใกล้เฟส B มากที่สุดก็จะถูกดูดเข้ามาแทน



รูปที่ 2-5 การทำงานของสเตปเปอร์มอเตอร์ ขั้นที่3

และเมื่อหยุดจ่ายกระแสให้เฟส B และจ่าย กระแสที่เฟส A แทน โรเตอร์ที่อยู่ใกล้เฟส A มากที่สุดจะถูกดูดเข้ามาแทน

จากหลักการนี้จะมีรูปแบบการจ่ายกระแสให้กับโรเตอร์ต่างๆเรียงตามลำดับ A B C ซึ่งจะอาศัยวงจรไอซีเข้ามาช่วย โดยส่งข้อมูลขนานในการส่งสัญญาณไปควบคุมสเตปเปอร์มอเตอร์ โดยจะส่งค่า 1,2, และ 4 ไปที่พอร์ต ตามลำดับวนไปเรื่อยๆ ซึ่งเป็นค่าในเลขฐานสอง

Step No.	Phase 1	Phase 2	Phase 3	Phase 4
1	ON	-	-	-
2	-	ON	-	-
3	-	-	ON	-
4	-	-	-	ON
5	ON	-	-	-
6	-	ON	-	-

ตารางที่ 2-1 ลอจิกการหมุนสเตปเปอร์มอเตอร์แบบ Half Step

0 0 0 0 0 0 0 1

0 0 0 0 0 0 1 0

0 0 0 0 0 1 0 0

รูปที่ 2-6 ลักษณะสัญญาณที่ส่งไปควบคุมสเตปเปอร์มอเตอร์ แบบ Half step

2.1.2.2 การควบคุมสเตปเปอร์มอเตอร์แบบสองเฟส หรือแบบฟูลสเตป (Full Step Drive)

การควบคุมสเตปเปอร์มอเตอร์แบบสองเฟสจะเป็นการส่งกระแสให้ขั้วของสเตเตอร์ทีละสองขั้วพร้อมๆกัน และเรียงลำดับกันไป เช่น ส่งกระแสไฟให้ขั้ว AB / BC / CD / DA / AB สลับกันไปเรื่อยๆ ซึ่งการควบคุมแบบฟูลสเต็ปนี้มีข้อดีตรงที่จะให้แรงบิดได้มากเป็นสองเท่าของแบบเวฟ แต่ก็จะใช้กระแสไฟมากกว่าเป็นสองเท่าเช่นเดียวกัน การควบคุมสเตปเปอร์มอเตอร์แบบฟูลสเต็ปมีลอจิกของการส่งกระแสไฟดังตารางข้างล่าง

Step No.	Phase 1	Phase 2	Phase 3	Phase 4
1	ON	ON	-	-
2	-	ON	ON	-
3	-	-	ON	ON
4	ON	-	-	ON
5	ON	ON	-	-
6	-	ON	ON	-

ตารางที่ 2-2 ลอจิกการหมุนสเตปเปอร์มอเตอร์แบบ Full Step

2.2 การสื่อสารแบบพาราเรลผ่านทางพรินเตอร์พอร์ต

พอร์ตพาราเรลมาตรฐานของ IBM 80x86 PC จะมีอยู่ทั้งหมด 25 ขาด้วยกัน แบ่งหน้าที่ของแต่ละพินออกได้ดังนี้

- **ขาส่งข้อมูลและกราวนด์ (Data lines and grounds)**

  ขาส่งข้อมูลพาราเรลจะมีทั้งหมด 8 ขา สามารถส่งข้อมูลได้ทีละหนึ่งไบต์ แบบซิมเพลกหรือฮาล์ฟดูเพลกก็ได้ (Simplex and Half-duplex transfer)

- **ขาแสดงสถานะพรินเตอร์ (Printer status signals)**

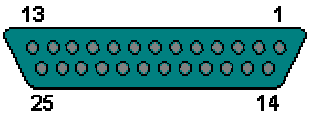
  ขาแสดงสถานะของพรินเตอร์จะมีทิศทางของสัญญาณจากพรินเตอร์ไปยัง PC มีอยู่ทั้งหมด 5 ขาด้วยกันคือ

  - **PE** (ขา 12) ใช้เมื่อกระดาษหมด
  - **BUSY** (ขา 11) ใช้เมื่อพรินเตอร์ยังไม่สามารถรับข้อมูลใหม่ๆได้
  - **/ERROR** (ขา 15) ใช้เมื่อพรินเตอร์เกิดขัดข้องขึ้น ไม่สามารถใช้งานได้
  - **SLCT** (ขา13) แสดงสัญญาณว่าพรินเตอร์เปิดอยู่และพร้อมใช้งาน
  - **/ACKNLG** (ขา10) สัญญาณตอบรับ ACK เพื่อขอรับไบต์ใหม่

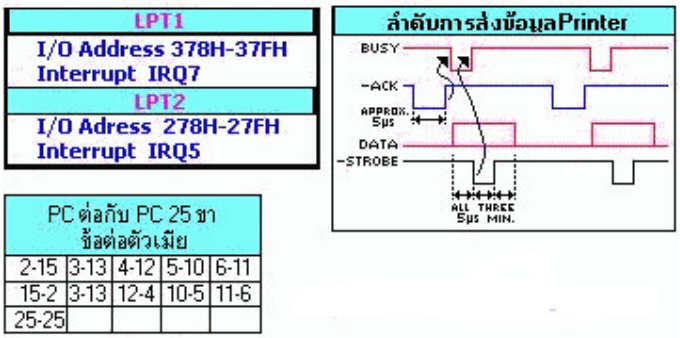
- **ขาควบคุมพรินเตอร์ (Printer control signals)**

  ขาควบคุมจะมีทิศทางการส่งข้อมูลจาก PC ไปยังพรินเตอร์ มีทั้งหมด 4 ขาด้วยกันคือ

  - **/STROBE** (ขา 1) ใช้ส่งสัญญาณถามสถานะพรินเตอร์
  - **/AUTOFD** (ขา 14) ใช้สั่งว่าให้ใส่กระดาษเองทีละแผ่นหรือไม่
  - **/INIT** (ขา 16) สัญญาณรีเซ็ต
  - **/SELIN** (ขา 17) พรินเตอร์จะรับข้อมูลเมื่อสัญญาณเป็น low เท่านั้น



รูปที่ 2-7 ขาของพอร์ตพาราเรลแบบ 25 ขา

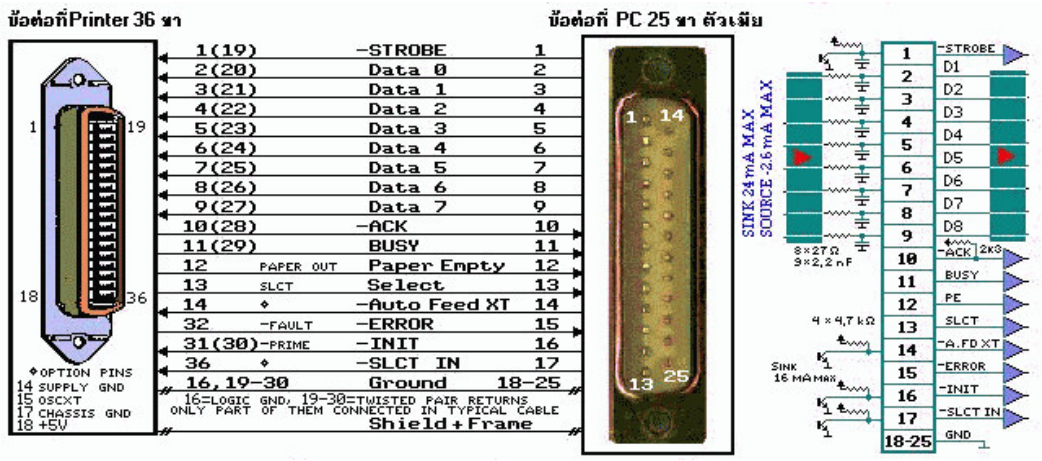


รูปที่ 2-8 แสดงแอดเดรสของพาราเรลพอร์ต และลำดับการส่งข้อมูลของพรินเตอร์

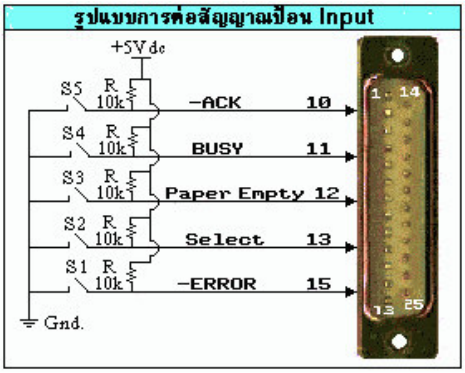
Pin	Name	Dir	Description
1	/STROBE	→	Strobe
2	D0	→	Data Bit 0
3	D1	→	Data Bit 1
4	D2	→	Data Bit 2
5	D3	→	Data Bit 3
6	D4	→	Data Bit 4
7	D5	→	Data Bit 5
8	D6	→	Data Bit 6
9	D7	→	Data Bit 7
10	/ACK	←	Acknowledge
11	BUSY	←	Busy
12	PE	←	Paper End
13	SEL	←	Select

Pin	Name	Dir	Description
14	/AUTOFD	→	Autofeed
15	/ERROR	←	Error
16	/INIT	→	Initialize
17	/SELIN	→	Select In
18	GND	—	Signal Ground
19	GND	—	Signal Ground
20	GND	—	Signal Ground
21	GND	—	Signal Ground
22	GND	—	Signal Ground
23	GND	—	Signal Ground
24	GND	—	Signal Ground
25	GND	—	Signal Ground

รูปที่ 2-9 หน้าที่และทิศทางของขาทั้งหมดของพาราแรลพอร์ต(ทิศทางออกจาก PC)



รูปที่ 2-10 อินเตอร์เฟสระหว่างข้อต่อที่พรินเตอร์และข้อต่อที่ PC



รูปที่ 2-11 รูปแบบการต่อสัญญาณป้อนกลับ

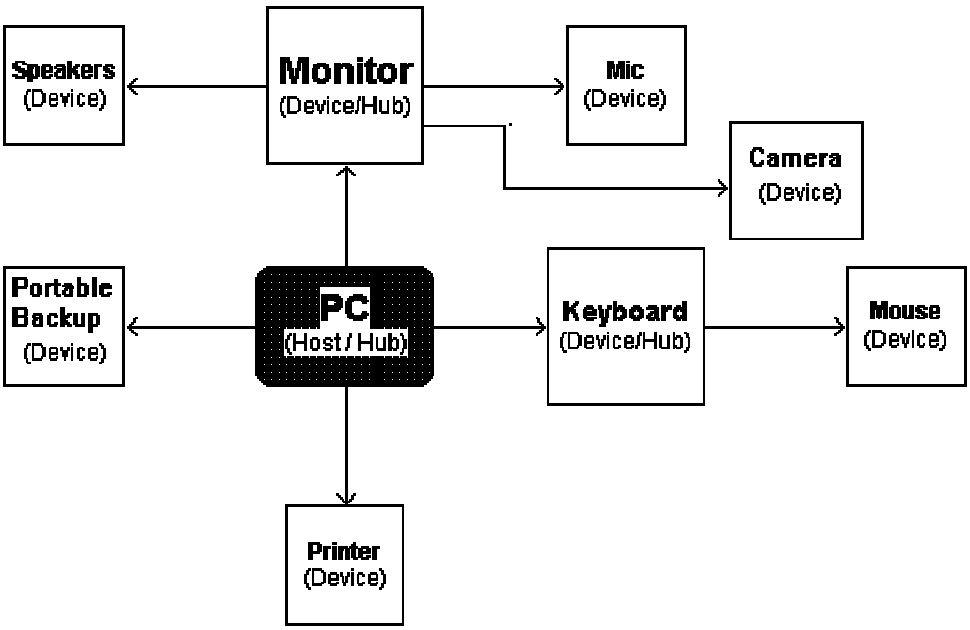
2.2.1 ตัวอย่างโปรแกรมการส่งข้อมูลออกพาราเรลพอร์ตด้วยภาษา C++

```
#include <dos.h>;
#include <iostream.h>;

int main(){
   int data, port;
   port=0x378;                    // LPT1
   data=0xAA;                     //10101010b
   outp(port,data);               // out AAh to LPT1
   cout<<"Out data "<< data<<" to port number "<<port<<endl;
   return 1;
}
```

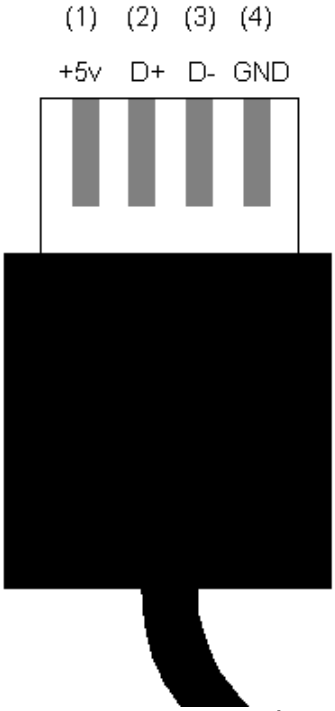
2.3 การสื่อสารแบบ USB

พอร์ตการสื่อสารแบบเก่าที่ใช้กันอย่างแพร่หลายคือ พาราเรลและซีเรียลพอร์ต (Parallel and Serial Port) มีความเร็วในการส่งข้อมูลที่ไม่มากนัก ความเร็วของพอร์ตพาราเรลอยู่ที่ 3200 กิโลบิตต่อวินาที ในขณะที่พอร์ตซีเรียลอยู่ที่ 115 กิโลบิตต่อวินาที และพอร์ตทั้งสองต้องการอินเทอรัพท์ของตัวเอง (IRQ:Interrupt) ซึ่งมีจำนวนจำกัดไม่สามารถเพิ่มจำนวนออกไปได้ และพอร์ตทั้งสองนี้ยังไม่รองรับพลั๊กแอนด์เพลย์ (Plug and Play) ซึ่งค่อนข้างยุ่งยากต่อผู้ใช้ที่ไม่ค่อยมีความรู้ทางด้านเทคนิค จึงได้มีการคิดค้นการสื่อสารด้วย USB (Universal Serial Bus) ขึ้นมา ซึ่งสามารถสื่อสารด้วยความเร็ว 12 เมกะบิตต่อวินาทีขึ้นมาโดยเน้นไปที่การสื่อสารทางด้านมัลติมีเดีย เช่นการสื่อสารกับฮาร์ดแวร์ที่แสดงรูปภาพและเสียง



รูปที่ 2-12 การอินเตอร์เฟส USB กับอุปกรณ์ต่อพ่วง

พอร์ต USB มีอยู่ 4 ขาด้วยกัน ขาแรกคือขา +5v(VCC), ขาที่สี่คือกราวนด์ ซึ่งอุปกรณ์ที่ใช้การสื่อสารแบบ USB นี้จะกินไฟไม่มาก อยู่ในระหว่าง 100mA ถึง 500mA ขาสองและสาม (D+,D-) เป็นขาที่ใช้ส่งสัญญาณแบบซีเรียล และเป็นอินเวอร์สซึ่งกันและกัน ซึ่งสามารถลดตัวรบกวนลงได้มากเมื่อต้องส่งสัญญาณไปในระยะทางไกล



รูปที่ 2-13 พอร์ต USB แสดงถึงหน้าที่ของขาสัญญาณ

ข้อมูลจะถูกส่งไปตามขา D+ และ D- โดยใช้การบีบอัดข้อมูลแบบ NRZI ด้วยความเร็ว 12 เมกะบิตต่อวินาทีด้วยสายที่มีการหุ้มตัวรบกวนได้ในระยะทางสูงสุด 5 เมตร ส่วนสายที่ไม่มีตัวหุ้มการรบกวนจะต้องมีความเร็วในการส่งน้อยลงเพื่อลดตัวรบกวน